IBATIS - RESULT MAPS

http://www.tutorialspoint.com/ibatis/ibatis_result_maps.htm

The resultMap element is the most important and powerful element in iBATIS. You can reduce upto 90% JDBC coding by using iBATIS ResultMap and in some cases allows you to do things that JDBC does not even support.

The design of the ResultMaps is such that simple statements don't require explicit result mappings at all, and more complex statements require no more than is absolutely necessary to describe the relationships.

This chapter would give you just a simple introduction of iBATIS ResultMap.

We have following EMPLOYEE table in MySQL:

```
CREATE TABLE EMPLOYEE (
   id INT NOT NULL auto_increment,
   first_name VARCHAR(20) default NULL,
   last_name  VARCHAR(20) default NULL,
   salary     INT  default NULL,
   PRIMARY KEY (id)
);
```

This table is having two records as follows:

```
mysql> select * from EMPLOYEE;
+----+------------+-----------+--------+
| id | first_name | last_name | salary |
+----+------------+-----------+--------+
|  1 | Zara       | Ali       |   5000 |
|  2 | Roma       | Ali       |   3000 |
+----+------------+-----------+--------+
2 row in set (0.00 sec)
```

Employee POJO Class:

To use iBATIS ResultMap, you do need to modify Employee.java file. So let us keep it as it is in last chapter.

```
public class Employee {
  private int id;
  private String first_name;
  private String last_name;
  private int salary;

  /* Define constructors for the Employee class. */
  public Employee() {}

  public Employee(String fname, String lname, int salary) {
    this.first_name = fname;
    this.last_name = lname;
    this.salary = salary;
  }

 /* Here are the required method definitions */
  public int getId() {
    return id;
  }
  public void setId(int id) {
    this.id = id;
  }
  public String getFirstName() {
    return first_name;
  }
  public void setFirstName(String fname) {
    this.first_name = fname;
  }
  public String getLastName() {
    return last_name;
  }
  public void setlastName(String lname) {
    this.last_name = lname;
  }
  public int getSalary() {
    return salary;
  }
  public void setSalary(int salary) {
    this.salary = salary;
  }

 } /* End of Employee */
```

Employee.xml File:

Here we would modify Employee.xml file to introduce <resultMap></resultMap> tag. This tag would have an id which is required to run this resultMap in our <select> tag's resultMap attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sqlMap
PUBLIC "-//ibatis.apache.org//DTD SQL Map 2.0//EN"
"http://ibatis.apache.org/dtd/sql-map-2.dtd">

<sqlMap namespace="Employee">

<!-- Perform Insert Operation -->
<insert >
   INSERT INTO EMPLOYEE(first_name, last_name, salary)
   values (#first_name#, #last_name#, #salary#)

   <selectKey resultClass="int" keyProperty="id">
      select last_insert_id() as id
   </selectKey>

</insert>

<!-- Perform Read Operation -->
<select >
   SELECT * FROM EMPLOYEE
</select>

<!-- Perform Update Operation -->
<update >
   UPDATE EMPLOYEE
   SET    first_name = #first_name#
   WHERE  id = #id#
</update>

<!-- Perform Delete Operation -->
<delete >
   DELETE FROM EMPLOYEE
   WHERE  id = #id#
</delete>

<!-- Using ResultMap -->
 <resultMap  >
    <result property="id" column="id"/>
    <result property="first_name" column="first_name"/>
    <result property="last_name" column="last_name"/>
    <result property="salary" column="salary"/>
</resultMap>
<select >
         SELECT * FROM EMPLOYEE
         WHERE id=#id#
</select>

</sqlMap>
```

IbatisResultMap.java File:

This file would have application level logic to read records from the Employee table using ResultMap:

```
import com.ibatis.common.resources.Resources;
import com.ibatis.sqlmap.client.SqlMapClient;
import com.ibatis.sqlmap.client.SqlMapClientBuilder;
import java.io.*;
import java.sql.SQLException;
import java.util.*;

public class IbatisResultMap{
  public static void main(String[] args)
   throws IOException,SQLException{
   Reader rd = Resources.getResourceAsReader("SqlMapConfig.xml");
   SqlMapClient smc = SqlMapClientBuilder.buildSqlMapClient(rd);

   int id = 1;
   System.out.println("Going to read record.....");
   Employee e = (Employee)smc.queryForObject
                ("Employee.useResultMap", id);

   System.out.println("ID:  " + e.getId());
   System.out.println("First Name:  " + e.getFirstName());
   System.out.println("Last Name:  " + e.getLastName());
   System.out.println("Salary:  " + e.getSalary());

   System.out.println("Record read Successfully ");

  }
}
```

Compilation and Run:

Here are the steps to compile and run the above mentioned software. Make sure you have set PATH and CLASSPATH appropriately before proceeding for the compilation and execution.

- Create Employee.xml as shown above.
- Create Employee.java as shown above and compile it.
- Create IbatisResultMap.java as shown above and compile it.
- Execute IbatisResultMap binary to run the program.

You would get following result which is a read operation on the EMPLOYEE table.

```
Going to read record.....
ID:  1
First Name:  Zara
Last Name:  Ali
Salary:  5000
Record read Successfully
```